

## CLAIMS

**1. (Currently Amended)** A method implemented in a component object-based multimedia framework of ~~for~~ an operating system executing on a computing device, the method comprising:

receiving a uniform resource locator (URL) as associated with one of a plurality of applications requesting media content;

identifying a scheme associated with the URL;

selecting a first object ~~operable~~ operated to handle the identified scheme associated with the URL to access parameter data from a location specified by the URL;

invoking the first object and determining whether the first object produces a second object according to a requested media content type;

~~based on the accessed parameter data, selecting a second object operable to read media content of a given type from the location specified by the URL wherein the second object comprises a media source object or a byte stream object;~~

in an event the second object is produced according to the requested media content type, then returning the second object to the one of the plurality of applications;

~~in an event where if the second object comprises a media source object, providing then returning the media source object to the one of the plurality of applications for generating media data in a~~ via a ~~component object-based pipeline, the pipeline comprising:~~

the media source object performing format-specific parsing to produce one or more streams of data that are sent through the pipeline, and

one or more media sinks rendering or archiving the media content ~~providing the media source object to the one of the plurality of applications the media content; and~~

~~in an event where if the second object comprises a byte stream object and the requested media content type is a media source object or a timeline object, selecting a then invoking one or more selected byte stream handlers to generate the requested media source object, wherein the byte stream handler is selected based in part on the byte stream object, and providing returning the selected requested media source object to the one of the plurality of applications for generating media data in a~~ via the ~~component object-based pipeline that allows for rendering or archiving the media content.~~

**2. (Currently Amended)** The method as recited in claim 1, wherein the selection of the second object is additionally based on information

contained in the URL indicating a type of multimedia data of ~~for~~ the media content.

**3. (Previously Presented)** The method as recited in claim 1, wherein the selection of the second object is additionally based on Multipurpose Internet Mail Extensions MIME data.

**4-6 (Canceled)**

**7. (Previously Presented)** The method as recited in claim 1, wherein the first object is produced using a scheme handler.

**8. (Previously Presented)** The method as recited in claim 1, wherein the second object is produced using a byte stream handler.

**9. (Previously Presented)** The method as recited in claim 1, wherein the first object is produced by a scheme handler and the second object is produced by a byte stream handler.

**10. (Previously Presented)** The method as recited in claim 1, wherein the first object is produced using a scheme handler selected from a list of two or more scheme handlers.

**11. (Previously Presented)** The method as recited in claim 1, wherein the second object is produced using a byte stream handler selected from a list of two or more byte stream handlers.

**12. (Previously Presented)** The method as recited in claim 1, wherein the first object is produced using a scheme handler selected from a list of two or more scheme handlers and the second object is produced using a byte stream handler selected from a list of two or more byte stream handlers.

**13. (Canceled)**

**14. (Previously Presented)** The method as recited in claim 1, wherein the second object is produced using a byte stream handler selected from a list of byte stream handlers and wherein each byte stream handler in the list has a selection value associated therewith.

**15. (Previously Presented)** The method as recited in claim 1, wherein the second object is produced using a byte stream handler selected from a list of byte stream handlers and wherein each byte stream handler in the list has a cost value associated therewith.

**16. (Previously Presented)** The method as recited in claim 1, wherein the second object is produced using a byte stream handler selected from a list of byte stream handlers and wherein each byte stream handler in the list has a cost value associated therewith, the cost value indicating how many bytes must be read by the byte stream handler in determining if the byte stream handler is appropriate for selecting the second object.

**17. (Currently Amended)** A computer-readable medium including computer-executable instructions for performing operations in a component object-based multimedia framework, the operations comprising:

receiving a uniform resource locator (URL) specifying a location of media content as associated with one of a plurality of applications requesting media content;

determining a scheme of the URL;

using the scheme to produce a byte stream object to handle the determined scheme associated with the URL to access parameter data;

using the byte stream object to generate a byte stream from the media content; and

using at least a portion of the byte stream to produce a media source object that,

the media source object performing format-specific parsing to produce one or more streams of data that are sent through the pipeline, and one or more media sinks rendering or archiving the media content

based on the accessed parameter data, reads the media content of a given type[,] from the location specified by the URL, and generates one or more streams of media data in a component object-based pipeline of the multimedia framework, the pipeline further comprising one or more media sinks ~~to allow for~~ rendering or archiving the media content.

**18. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of using the byte stream additionally includes using a file extension indicated in the URL to select the source object.

**19. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a byte stream object includes choosing a scheme handler and using the chosen scheme handler to produce the byte stream object.

**20. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a byte stream object includes choosing a scheme handler from a list of scheme handlers and using the chosen scheme handler to produce the byte stream object.

**21. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes choosing a byte stream handler and using the chosen byte stream handler to produce the media source object.

**22. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes choosing a byte stream handler from a list of byte stream handlers and using the chosen byte stream handler to produce the media source object.

**23. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes choosing a byte stream handler from a list of byte stream handlers and using the chosen byte stream handler to produce the media source object and

wherein the list of byte stream handlers is ordered based on a selection values associated with the byte stream handlers.

**24. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a source object includes choosing a byte stream handler from a list of byte stream handlers and using the chosen byte stream handler to produce the media source object and wherein the list of byte stream handlers is ordered based on a cost values associated with the byte stream handlers.

**25. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes choosing a byte stream handler from a list of byte stream handlers and using the chosen byte stream handler to produce the media source object and wherein and each byte stream handler in the list has a cost value associated therewith, the cost value indicating an amount of data that must be read by the byte stream handler in determining if the byte stream handler is appropriate for producing the media source object.

**26. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object



includes using a look-up process to select a subset of byte stream handlers from a set of byte stream handlers and using one of the subset of byte stream handlers to produce the media source object.

**27. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a source object includes using a look-up process to:

select a number of byte stream handlers; and

invoke the selected byte stream handlers one at a time until a byte stream handler produces a media source object.

**28. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes using a look-up process to:

select a number of byte stream handlers; and

invoke the selected byte stream handlers one at a time in a predetermined order until a byte stream handler produces a media source object.

**29. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes using a look-up process to:

select a number of byte stream handlers; and

invoke the byte stream handlers one at a time in a predetermined order based on cost values associated with the selected byte stream handlers until a byte stream handler produces a media source object.

**30. (Currently Amended)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes using a look-up process to:

select a number of byte stream handlers;

compile ~~compiling~~ a first list of byte stream handlers, each of the byte stream handlers in the first list being associated with the type of the media content;

compile ~~compiling~~ a second list of byte stream handlers, each of the byte stream handlers in the second list not being associated with the type of the media content;

invoke the byte stream handlers in the first list one at a time until either a byte stream handler in the first list produces a media source object or until all byte stream handlers in the first list have been invoked without producing a media source object; and

in an event if all byte stream handlers in the first list have been invoked and none of the invoked byte stream handler from the first list produced a media

source object, then invoking each of the byte stream handlers in the second list one at a time until either a either a byte stream handler from the second list produces a source object or until all byte stream handlers in the second list have invoked without producing a media source object.

**31. (Currently Amended)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes using a look-up process to:

select a number of byte stream handlers;

compile ~~compiling~~ a first list of byte stream handlers, each of the byte stream handlers in the first list being associated with the type of the media content, the byte stream handlers in the first list being ordered according to cost values associated with the byte stream handlers in the first list;

compile ~~compiling~~ a second list of byte stream handlers, each of the byte stream handlers in the second list not being associated with the type of the media content;

invoke the byte stream handlers in the first list one at a time in order until either a byte stream handler in the first list produces a media source object or until all byte stream handlers in the first list have been invoked without producing a media source object; and

in an event if all byte stream handlers in the first list have been invoked and none of the invoked byte stream handler from the first list produced a source object, then invoking each of the byte stream handlers in the second list one at a time until either a either a byte stream handler from the second list produces a media source object or until all byte stream handlers in the second list have invoked without producing a media source object.

**32. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a byte stream object includes using a look-up process to:

select a number of scheme handlers; and

invoke the scheme handlers in the list one at a time until a scheme handler produces a byte stream object.

**33. (Previously Presented)** The computer-readable medium as recited in claim 17, wherein the operation of producing a byte stream object includes using a look-up process to:

select a number of scheme handlers; and

invoke the scheme handlers in the list one at a time in a predetermined order until a scheme handler produces a byte stream object.

**34. (Currently Amended)** The computer-readable medium as recited in claim 17, wherein the operation of producing a media source object includes using a look-up process to:

select a number of scheme handlers;

compile ~~compiling~~ a list of scheme handlers, each of the scheme handlers in the list of scheme handlers being associated with the scheme of the URL;

invoke the scheme handlers in the list of scheme handlers one at a time until either a byte stream object is produced, a media source object is produced, or all scheme handlers in the list of scheme handlers have been invoked and neither a byte stream object nor a media source object have been produced;

in an event if either a media source object or a byte stream object has been produced, then determining if an application has requested a media source object;

in an event if an application has requested a media source object, then providing the media source object to the application; and

in an event if the application has not requested a media source object, then compile ~~compiling~~ a list of byte stream handlers[,] and invoking the byte stream handlers in the first list one at a time until either a byte stream handler in the list produces a media source object or until all byte stream handlers in the first list have been invoked without producing a media source object.

**35. (Currently Amended)** A computerized system including:  
an object selection module in a component object-based multimedia framework operable to:

receive a uniform resource locator (URL) as associated with one of a plurality of applications requesting media content;

determine a scheme of the URL specifying a location of media content;

use the scheme to produce a byte stream object to handle the determined scheme associated with the URL to access parameter data;

use the byte stream object to generate a byte stream from the media content; and

use a portion of the byte stream to produce a media source object that, based on the accessed parameter data, reads the media content of a given type[,] from the location specified by the URL, and generates one or more streams of media data in a component object-based pipeline of the multimedia framework, the pipeline further comprising to allow for one or more media sinks rendering or archiving the media content.

**36. (Previously Presented)** The computerized system as recited in claim 35, wherein the byte stream object is produced using a scheme handler.

**37. (Previously Presented)** The computerized system as recited in claim 35, wherein the media source object is produced using a byte stream handler.

**38. (Previously Presented)** The computerized system as recited in claim 35, wherein the byte stream object is produced using a scheme handler that is selected from a list of scheme handlers, the list being selected based on the scheme of the URL.

**39. (Previously Presented)** The computerized system as recited in claim 35, wherein the media source object is produced using a byte stream handler that is selected from a list of byte stream handlers, the list being selected based on a byte stream generated from data at the location indicated by the URL and a portion of the URL.

**40. (Previously Presented)** The computerized system as recited in claim 35, wherein the byte stream object is produced using a scheme handler that is selected from a list of scheme handlers, the list being selected based on the scheme of the URL and ordered based on cost values associated with each of the scheme handlers in the list.

**41. (Previously Presented)** The computerized system as recited in claim 35, wherein the operation of producing a media source object includes using a look-up process to:

select a number of byte stream handlers; and

invoke the selected byte stream handlers one at a time until a byte stream handler produces a media source object.

**42. (Previously Presented)** The computerized system as recited in claim 35, wherein the operation of producing a byte stream object includes using a look-up process to:

select a number of scheme handlers; and

invoke the scheme handlers in the list one at a time until a scheme handler produces a byte stream object.

**43. (Currently Amended)** A system of ~~for~~ a component object-based multimedia framework comprising:

means for receiving a uniform resource locator (URL) as associated with one of a plurality of applications requesting media content;

means for identifying a scheme associated with the URL;

means for selecting a scheme handler based on the identified scheme, the scheme handler producing a byte stream object operable to handle the identified



scheme associated with the URL to access parameter data from the location pointed to by the URL and produce a byte stream from the accessed parameter data; and

means for selecting a byte stream handler based on the accessed parameter data, the byte stream handler producing a media source object operable to read the media content of a given type, from the location specified by the URL, and ~~operable~~ operated to generate data in a component object-based pipeline, the pipeline further comprising one or more media sinks ~~to allow~~ ~~for~~ rendering or archiving the media content.

**44. (Previously Presented)** The system as defined in claim 43, further comprising a lookup means for producing a list of scheme handlers, wherein the means for selecting the scheme handler selects the scheme handler from the list of scheme handlers.

**45. (Previously Presented)** The system as defined in claim 43, further comprising a lookup means for producing a list of byte stream handlers, wherein the means for selecting the byte stream handler selects the byte stream handler from the list of byte stream handlers.

**46. (Previously Presented)** The system as defined in claim 43, wherein the means for selecting a scheme handler selects the scheme handler in response to a request from an application.

**47. (Previously Presented)** The system as defined in claim 43, wherein the means for selecting a byte stream handler selects the byte stream handler in response to a request from an application.

**48. (Previously Presented)** The system as defined in claim 43, wherein the media source object produced by the byte stream handler is employed as a component in a multi-component media processing pipeline.

**49. (Previously Presented)** The system as defined in claim 43, wherein the media source object produced by the byte stream handler is employed as a component in a multi-component media processing pipeline in a media engine.

**50. (Previously Presented)** The system as defined in claim 43, wherein the means for selecting a scheme handler selects the scheme handler in response to a request from an application and wherein the media source object produced by the byte stream handler is employed as a component in a multi-

component media processing pipeline in a media processing module that is an operational module in a operating system.

**51. (Previously Presented)** The system as defined in claim 43, wherein the means for selecting a byte stream handler employs a lookup module.

**52. (Currently Amended)** One or more computer-readable storage media having comprising computer-executable instructions encoded thereon, the computer-executable instructions, upon execution configuring a processing unit to perform ~~for performing~~ operations comprising:

responsive to receiving a uniform resource locator (URL) for media content from one of a plurality of applications executing on an operating system of a computing device, selecting a media source object using an object-based multimedia framework of the operating system, wherein the selecting comprises identifying a scheme associated with the URL[[,]];

identifying one or more available scheme handler objects operable to read information associated with the scheme of the URL;

selecting invoking the [[a]] scheme handler object operable to handle the identified scheme associated with the URL to access parameter data from a location specified by the URL, and

based on the accessed parameter data, ~~selecting~~ generating a selected media source object; and

providing the selected media source object to the one of the plurality of applications executing on the operating system of the computing device via ~~to~~ ~~thereby allow for a~~ an object-based media pipeline that comprises the media source object ~~configured to generate~~ that performs format specific parsing to produce one or more streams of media data and providing the one or more streams of media data to a media sink object, the media sink object ~~configured to render~~ rendering or ~~archive~~ archiving the media content.